Improvements in the Index-Calculus algorithm for Solving the Discrete Logarithm problem over $\mathbb{F}_p$

Raminder Ruprai

October 29, 2007
Summary

- The Discrete Logarithm Problem
- Preliminaries: Smoothness & The Subexponential Function
- ‘Original’ Index-Calculus Algorithm
- Worked Example
- The Number Field Sieve Improvements
The Discrete Logarithm Problem (DLP) in general is as follows: -
Given a group $G$ with a binary operation written multiplicatively, the DLP is, given $a, b \in G$ where $b \in \langle a \rangle$, find an integer $x$ such that

$$a^x = b.$$
In a finite field of prime order..
For $t, y \in \mathbb{N}$, we say $t$ is $y$-smooth, if all the primes in the prime factorisation of $t$ are $\leq y$.
For example: $100 = 2^2 \times 5^2$ so we say 100 is 5-smooth.
The subexponential function can be used to describe the running time of algorithms which are slower than polynomial but faster than exponential. Given an algorithm $M$ with argument $p$, we say that $M$ completes in:

- polynomial time if the expected running time is $O((\log p)^c)$,
- exponential time if the expected running time is $O(p^c)$,

for some positive $c$. 
The subexponential function

The subexponential function is then

$$L_n[v; c] = e^{(c+o(1))(\log n)^v(\log \log n)^{1-v}}.$$

- When $v = 1$ we have $n^c$ i.e. exponential time
- When $v = 0$ we have $(\log n)^c$ i.e. polynomial time

So the closer we head towards $v = 0$ the faster our algorithm is. Informally we let $c$ be a general constant and $n$ be the known argument, in this case our big prime $p$, so we just denote the subexponential function as $L(v)$. 
A Factor base $\mathcal{F}$ is chosen consisting of small primes e.g. $\mathcal{F} = [2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots]$.

A matrix $M$ is created so that the columns represent the discrete logs to the base $a$ ($\log_a$) of elements of $\mathcal{F}$. The rows of the matrix will represent relations which we now need to find/construct.
Generating Relations

- Pick a random integer $v \in [1..p - 1]$ and compute
  \[ c \equiv a^v \pmod{p} \]

- $c$ is an element of $\mathbb{Z}_p^*$ but we treat it as an integer and factorise it. If all the factors of $c$ are in $\mathcal{F}$ then we say that $c$ is $\mathcal{F}$-smooth.

- Given that $c = p_1^{b_1} \cdot p_2^{b_2} \cdot \ldots \cdot p_n^{b_n}$ where all the $p_i \in \mathcal{F}$ then
  \[ v \equiv b_1 \log_a p_1 + b_2 \log_a p_2 + \ldots + b_n \log_a p_n \pmod{p - 1}. \]
Filling in the Matrix & Solving

- From this relation we then add a sparse row to the matrix $M$ which has entries $b_i$ in the column representing $\log_a p_i$ and 0’s elsewhere. In addition we add $v$ to the solution vector $S$.
- We want to solve $MX = S$ for $X$, where $X$ is a vector whose elements are $\log_a p_i$. This is solved over $p - 1$ which is not prime so some elements will not have inverses.
- Therefore we continue finding relations until we have more rows that columns i.e. more relations that elements in the factor base.
- When enough relations are found we solve for the discrete logs of the elements in the factor base.
Solving for DLP

- We now know $\log_a p_i \forall p_i \in \mathcal{F}$
- Then we take random $j$’s until $a^j b$ is $\mathcal{F}$-smooth. Therefore we can write

$$a^j b \equiv p_1^{d_1} p_2^{d_2} \ldots p_n^{d_n} \pmod{p}$$

for $p_i \in \mathcal{F}$. Then we get

$$j + \log_a b \equiv d_1 \log_a p_1 + d_2 \log_a p_2 + \ldots + d_n \log_a p_n.$$ 

- We know $j$, all the $d_i$’s and all the discrete logs of the elements of the factor base so we can re-arrange the equation to find $\log_a b$. 

Raminder Ruprai

Improvements in the Index-Calculus algorithm for Solving the DLP
Example with $p = 1009$

Given a prime $p = 1009$ and base element $11$ we want to find $x$ such that

$$11^x \equiv 946 \pmod{1009}$$

- The size of the factor base should be around $L(1/2)$ for this algorithm so take our factor base to be $\mathcal{F} = \{2, 3, 5, 7, 11, 13\}$.
- We need at least 6 relations but we have 1 already

$$\log_{11} 11 \equiv 1 \pmod{1008}$$

We continue by choosing random powers of 11 mod 1009 and see whether they are $\mathcal{F}$-smooth.
We get the following

\[
11^{300} = 112 \mod 1009 \quad \text{which factorises as} \quad 2^4 \times 7,
\]
\[
11^{23} = 390 \mod 1009 \quad \text{which factorises as} \quad 2 \times 3 \times 5 \times 13,
\]
\[
11^{900} = 400 \mod 1009 \quad \text{which factorises as} \quad 2^4 \times 5^2,
\]
\[
11^{134} = 768 \mod 1009 \quad \text{which factorises as} \quad 2^8 \times 3,
\]
\[
11^{797} = 165 \mod 1009 \quad \text{which factorises as} \quad 3 \times 5 \times 11,
\]
\[
11^{43} = 65 \mod 1009 \quad \text{which factorises as} \quad 5 \times 13.
\]
These can then be transformed into the associated congruences as follows,

\[
\begin{align*}
\log_{11} 11 & \equiv 1 \pmod{1008}, \\
4 \log_{11} 2 + \log_{11} 7 & \equiv 300 \pmod{1008}, \\
\log_{11} 2 + \log_{11} 3 + \log_{11} 5 + \log_{11} 13 & \equiv 23 \pmod{1008}, \\
4 \log_{11} 2 + 2 \log_{11} 5 & \equiv 900 \pmod{1008}, \\
8 \log_{11} 2 + \log_{11} 3 & \equiv 134 \pmod{1008}, \\
\log_{11} 3 + \log_{11} 5 + \log_{11} 11 & \equiv 797 \pmod{1008}, \\
\log_{11} 5 + \log_{11} 13 & \equiv 43 \pmod{1008}.
\end{align*}
\]
The matrix, $M$, that we get from these congruences is

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 \\
4 & 0 & 0 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 \\
4 & 0 & 2 & 0 & 0 & 0 \\
8 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1
\end{pmatrix}
$$

We also get a solution vector

$$
S = (1, 300, 23, 900, 134, 797, 43)^T.
$$
Example cont.5

Solving $MX = S$ for $X$ we get the following discrete logs:

\[
\begin{align*}
\log_{11} 2 &= 886, \\
\log_{11} 3 &= 102, \\
\log_{11} 5 &= 694, \\
\log_{11} 7 &= 788, \\
\log_{11} 11 &= 1, \\
\log_{11} 13 &= 357.
\end{align*}
\]

We now have to attempt to find a power, $\gamma$, of 11 such that $946 \times 11^\gamma$ is $\mathcal{F}$-smooth. After a few attempts we get $\gamma = 491$ and therefore:

\[
946 \times 11^{491} \equiv 624 = 2^4 \times 3 \times 13 \pmod{1009}
\]

Going back to our original DLP, $11^x \equiv 946$, then we have that

\[
x \equiv 4 \log_{11} 2 + \log_{11} 3 + \log_{11} 13 - 491 \equiv 488
\]

So the solution to the DLP is $x = 488$. 
Proposition

Given a number \( n \leq L(s) \). The probability that \( n \) is \( L(t) \)-smooth is \( L(t - s) \).

- So to obtain enough relations to solve the linear algebra is the reciprocal of that, \( L(s - t) \). What does that mean here?
- \( c = a^y \) can have any value in \( \mathbb{Z}_p^* \) so has order of magnitude \( p \) or \( L(1) \). To minimise the running time of the algorithm the factor base has to have size \( L(1/2) \) therefore the probability that a \( c \) is \( \mathcal{F} \)-smooth is \( L(1/2 - 1) \).
- So to generate enough relations we get a running time of \( O(L(1/2)) \).
The Number Field Sieve

- In the Number Field Sieve Index-Calculus Algorithm relations are constructed in a different way.
- Motivation: In order to reduce the running time we attempt to reduce the size of the number we are factorising.
- If we could restrict the magnitude of $a^\nu$ to $L(2/3)$ then we could reduce the running time of the algorithm. But this is hard.
A Different way to get relations

Instead we do the following.

- Choose a polynomial $f$ with root $\alpha \in \mathbb{C}$ and such that $f(m) \equiv 0 \pmod{p}$ for some integer $m$. There is a homomorphism $\psi : \alpha \rightarrow m \pmod{p}$.
- The factor base $\mathcal{F}$ will have two parts $\mathcal{F}_Q$ (consisting of small prime numbers) and $\mathcal{F}_K$ (consisting of degree 1 prime ideals).
- Now we choose $c, d \leq L(1/3)$ and consider $c + dm$. For every $c + dm$ there is an ideal $(c + d\alpha)$ associated with it.
Different way to get relations

Thus

\[
\begin{align*}
\text{LEFT} & \equiv \text{RIGHT} \\
c + dm & \equiv \psi(c + d\alpha) \\
\prod_{p_i \in \mathcal{F}_Q} p_i^{e_i} & \equiv \psi(\prod_{p_i \in \mathcal{F}_K} p_i^{f_i})
\end{align*}
\]
Advantage, Quicker...or another problem?

- So far, ok, but there looks to be a problem...
- What do we do with the ‘stuff’ on the right.
- As the prime factorisation of the ideal \((c + d\alpha)\) is different from the prime factorisation of the number \(c + dm\) with enough relations we can cancel out all the prime ideals on the right to get 1.
- With the number field, this is not a trivial issue as getting 1 on the right is not as straightforward as just cancelling.
With the number field out of the way...

- From the previous slide we have the following

  \[ \prod_{p_i \in \mathcal{F}_Q} p_i^{e_i} \equiv 1 \pmod{p}. \]

- Taking the logarithm to the base \(a\) of both sides we have

  \[ \sum_{p_i \in \mathcal{F}_Q} e_i \log_a p_i \equiv 0 \pmod{p - 1}. \]

  With enough relations we can use linear algebra to solve for the discrete logarithm of the primes in the factor base.

  However, we still do not have \(\log_a b\).
Solving the DLP for a specific $b$

In the Number field sieve algorithm this is not as trivial as before. Roughly speaking we have to do the following:

- We choose an $l \in [1, p - 1]$ such that
  \[ a^l b \equiv \prod_{j} q_j \pmod{p}, \]
  where the $q_j$'s are primes of 'medium' size but do not have to be distinct.
- For each $q_j$, we need to set up a new factor base of prime ideals over a different extension field.
- We then need to find enough relations to cancel out all the prime ideals in order to find $\log_a q_j$ as a sum of discrete logs of elements in $\mathcal{F}_Q$. We know these from the previous slides so we now know $\log_a b$!
So to conclude the running time is...

The running time using the Number Field Sieve as described by Daniel Gordon of the University of Georgia in 1992 is

$$O \left( L \left( \frac{1}{3}, 3^{\frac{2}{3}} \right) \right).$$

When the DLP is over different types of finite fields such as $\mathbb{F}_{p^m}$ the constant can be improved. The crossover point between the 'Original' Index-Calculus Algorithm and the Number Field Sieve is 218 bits so one could say that this algorithm is unpractical for reasonable numbers. Then again you wouldn’t be trying to solve an easy DLP...
Thank you
Any questions?